Why I’m Excited to work in Computing Division in “10 minutes”

Maya Wospakrik on behalf of FNAL Computing Division

Fermilab

A little bit about myself

01 Postdoc at Scientific Computing Division (SCD)

I work mostly in MicroBooNE and ICARUS experiments. Also an SBN collaborator

02 Neutrino Physicist

Thesis work on MINERvA experiment on cross-section analysis using machine learning.

Now working mostly on non-standard oscillation analysis within SBN experiments.

03 Member of SCIDAC-4-HEP Data Analytics for HPC

Researching Big Data and HPC tools to parallelize existing softwares used for the SBN neutrino oscillation analysis

A little bit about myself

01 Postdoc at Scientific Computing Division (SCD)

I work mostly in MicroBooNE and ICARUS experiments. Also an SBN collaborator

02 Neutrino Physicist

Thesis work on MINERvA experiment on cross-section analysis using machine learning.
Now working mostly on non-standard oscillation analysis within SBN experiments.

03 Member of SCIDAC-4-HEP Data Analytics for HPC

Researching Big Data and HPC tools to parallelize existing tools used for the SBN neutrino oscillation analysis

A little bit about myself

01 Postdoc at Scientific Computing Division (SCD)

I work mostly in MicroBooNE and ICARUS experiments. Also an SBN collaborator

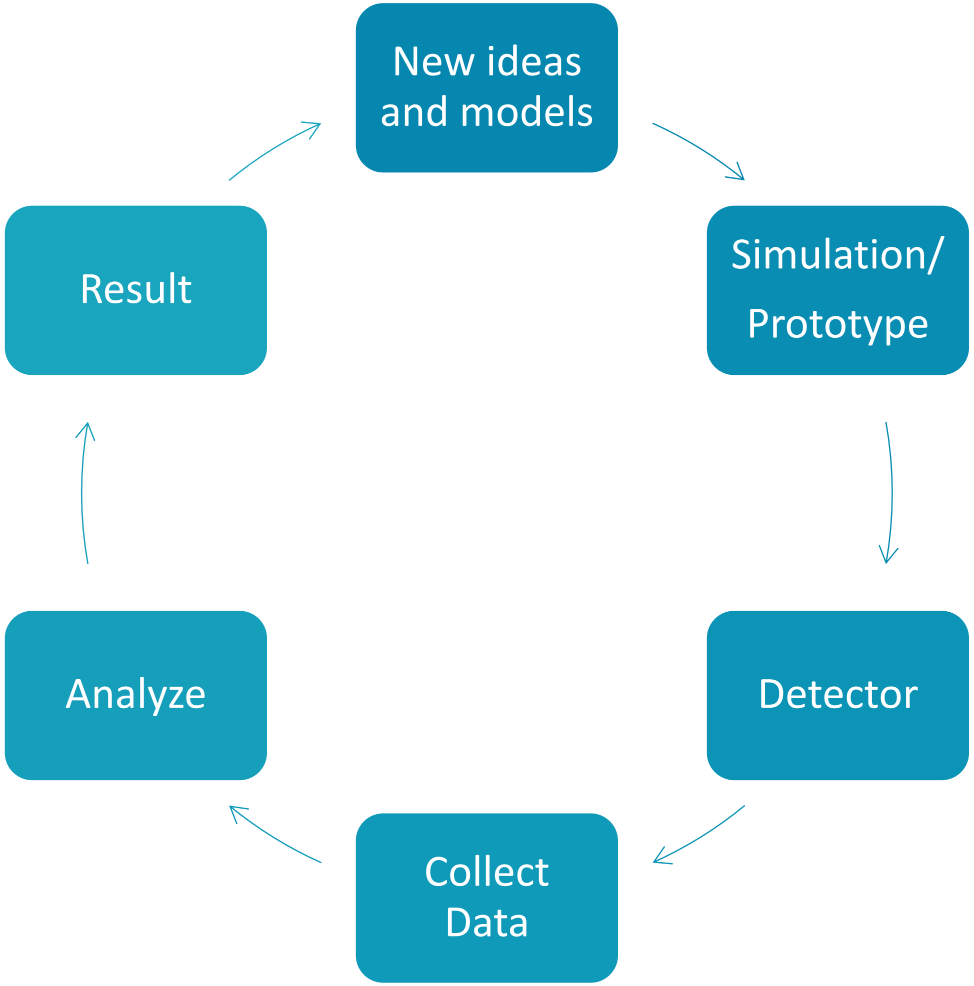
02 Neutrino Physicist

Thesis work on MINERvA experiment on cross-section analysis using machine learning. Now working mostly on non-standard oscillation analysis within SBN experiments.

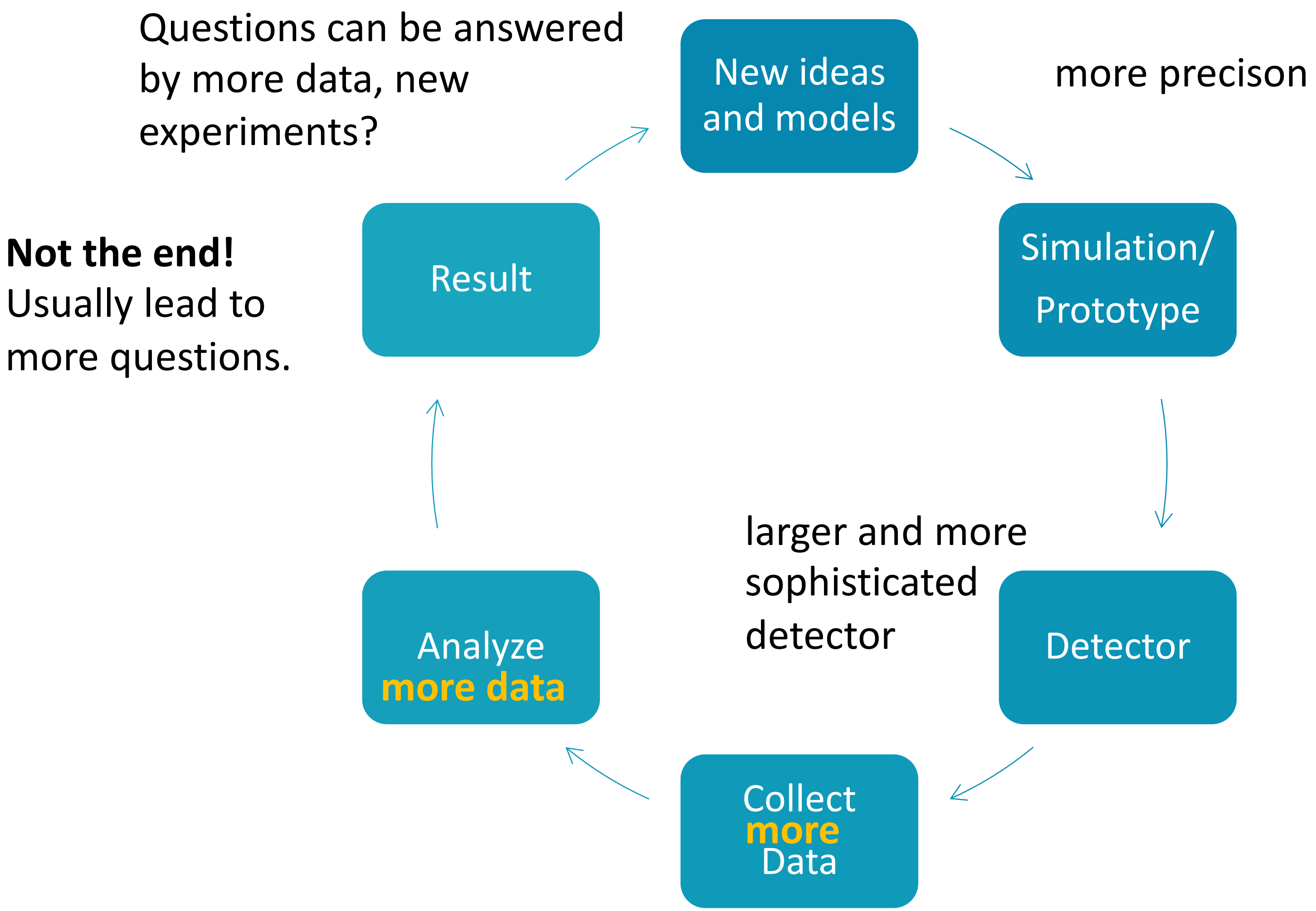
03 Member of SCIDAC-4-HEP Data Analytics for HPC

Researching Big Data and HPC tools to parallelize existing tools used for the SBN neutrino oscillation analysis

The Road to Physics Discovery



The Road to Physics Discovery



The Road to Physics Discovery

Questions can be answered by more data, new experiments?

New ideas and models

more precison

Simulation/ Prototype

larger and more sophisticated detector

Detector

Collect **more** Data

Analyze **more data**

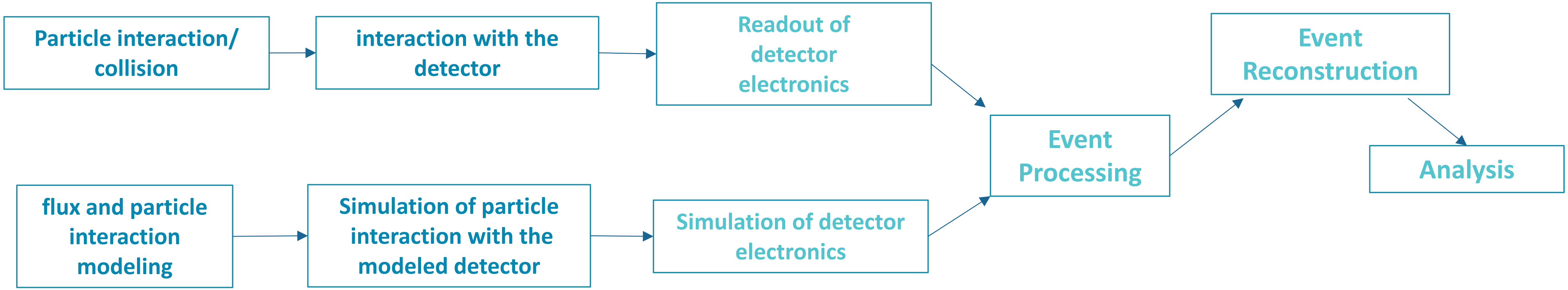
Result

Not the end!
Usually lead to more questions.

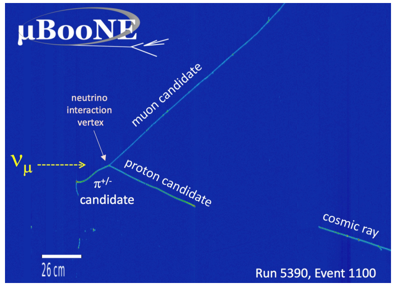
#dark #netflix

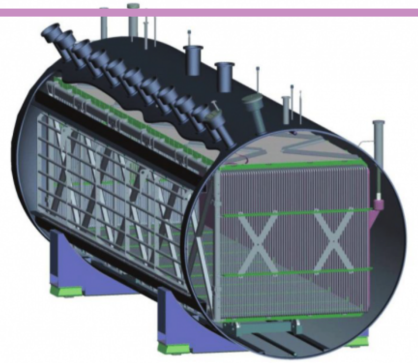
push the frontiers of accelerator and detector technology, bringing **enormous challenges** to software and computing

Typical Workflow in Experimental Physics



Big Data Challenges



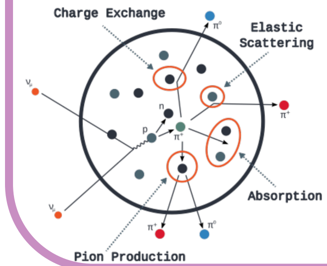


Particle interaction/ collision → interaction with the detector → Readout of detector electronics

Data Acquisition

Fast Machine Learning to enable separation of signal from a large noisy background
And reduce data volume

flux and particle interaction modeling → Simulation of particle interaction with the modeled detector → Simulation of detector electronics



GEANT4
A SIMULATION TOOLKIT

LArSoft

Simulation

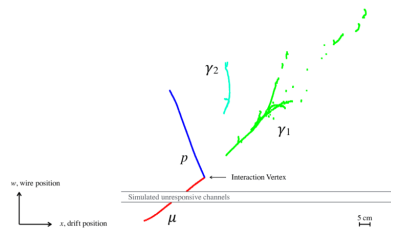
More precision leads to more computation.
Computationally expensive, but can be optimized for parallelization.
(see talk from J. Isaacson, "Event Generation with GPUs")

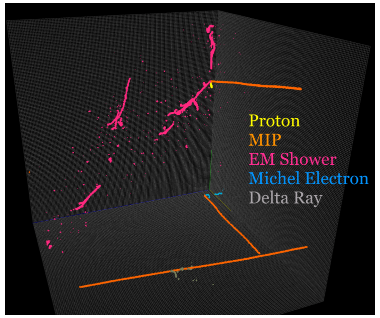
Big Data Challenges

Pattern-recognition reconstruction, Machine Learning

Reduce time to reconstruct events:

- vectorization, parallelization and code portability to GPUs.
- Convolutional neural networks which is becoming critical analysis techniques in HEP





Event Reconstruction

Event Processing

HPC Facilities

Higher volumes of data requires more processing unit. Leads to use of HEPCloud and Theta Supercomputer at Argon National Laboratory (ANL)

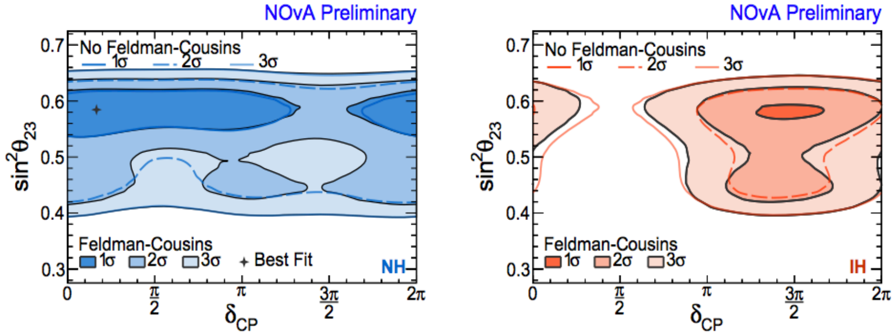
Big Data Challenges

[EPJ Web Conf., 214, 05012 (2019)]

Analysis – shorten “time to physics”

Accelerating NOvA’s complex computational antineutrino appearance/disappearance analysis on Cori supercomputer at **National Energy Research Scientific Computing Center (NERSC), Lawrence Berkley National Laboratory.**

Reduce the time required to analyze the data from scale of weeks to days.



Analysis

Big Data Challenges

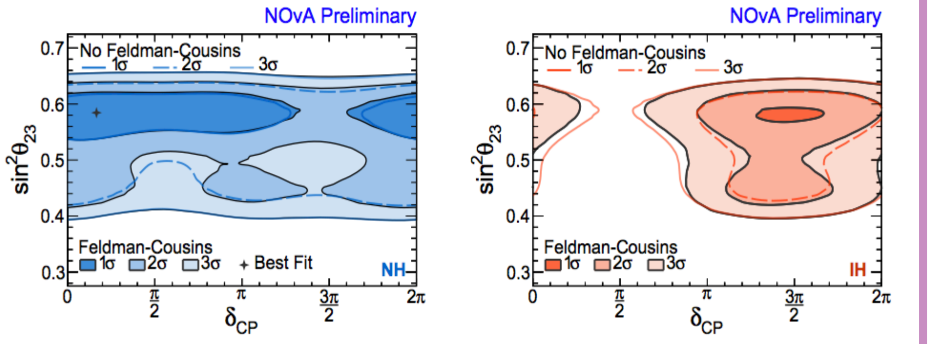
Analysis – shorten “time to physics”

Accelerating NOvA’s complex computational antineutrino appearance/disappearance analysis on Cori supercomputer at **National Energy Research Scientific Computing Center (NERSC), Lawrence Berkley National Laboratory.**

Reduce the time required to analyze the data from scale of weeks to days.

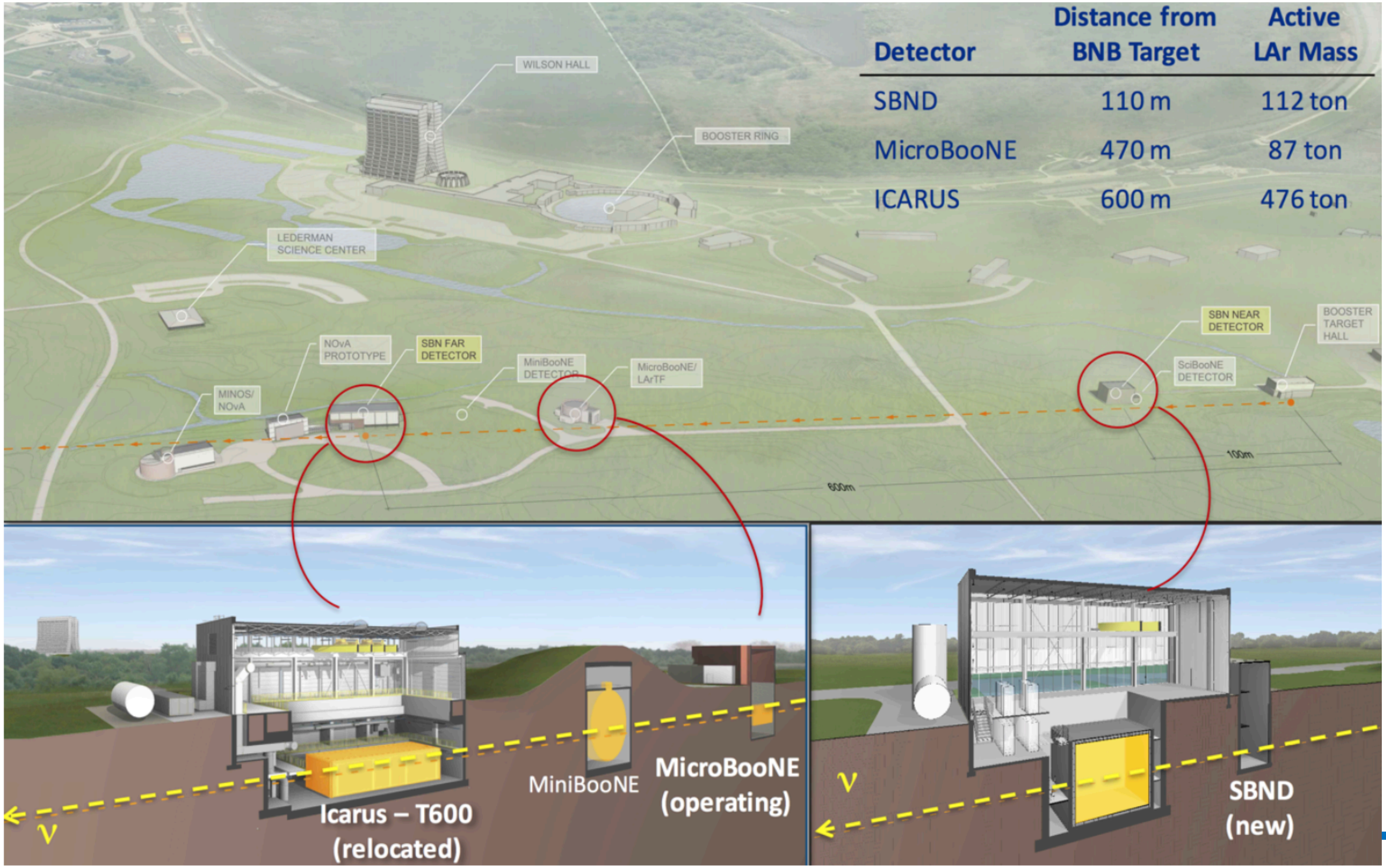
<u>My research interest</u>:
learn from NOvA experiences and accelerate the tools for SBN oscillation analysis in HPC environment: Cori supercomputing at NERSC

[EPJ Web Conf., 214, 05012 (2019)]



Analysis

SBN Program



Detector	Distance from BNB Target	Active LAr Mass
SBND	110 m	112 ton
MicroBooNE	470 m	87 ton
ICARUS	600 m	476 ton

- **SBN**: **3 liquid argon detectors** located in the **Fermilab Booster Neutrino Beamline** with main goal to search for short baseline anomaly.
  - See talks from D. Barker “SBND in 10 minutes” and J. Mueller “Solving the Sterile Neutrino Puzzle with ICARUS” earlier today!
- Each detector shares the **same neutrino flux** and **argon cross-sections** -> measurement is **highly correlated -> reduce systematics in oscillation analysis**

χ^2 Calculation for υ Oscillation Sensitivity

- Benchmarking:
  - SBNfit framework: one of the fitting framework used by SBN collaboration (Physics Group at Nevis Lab at Columbia University)
- Sensitivity is calculated by computing a χ^2 surface in the $(\Delta m_{41}^2, \sin^2(2\theta))$ oscillation parameter plane:

$$\chi^2(\Delta m_{41}^2, \sin^2 2\theta) = \sum_{i,j} \left[N_i^{null} - N_i^{osc}(\Delta m_{41}^2, \sin^2 2\theta)\right] (E_{ij})^{-1} \left[N_j^{null} - N_j^{osc}(\Delta m_{41}^2, \sin^2 2\theta)\right]$$

Fermilab

χ^2 Calculation for υ Oscillation Sensitivity

Sensitivity is calculated by computing a χ^2 surface in the $(\Delta m^2_{41}, \sin^2(2\theta))$ oscillation parameter plane:

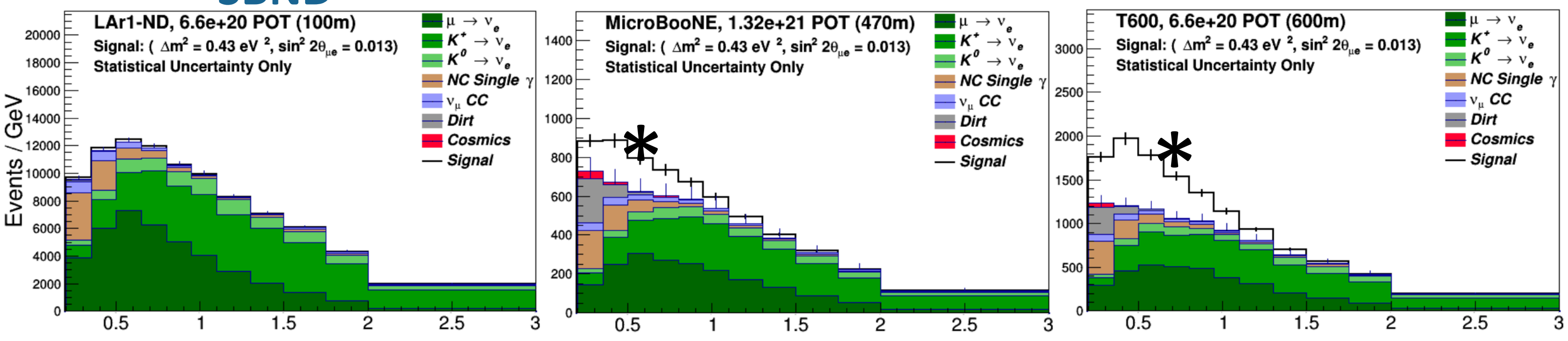
$$\chi^2(\Delta m^2_{41}, \sin^2 2\theta) = \sum_{i,j} \left[N_i^{null} - N_i^{osc*}(\Delta m^2_{41}, \sin^2 2\theta)\right] (E_{ij})^{-1} \left[N_j^{null} - N_j^{osc}(\Delta m^2_{41}, \sin^2 2\theta)\right]$$

$$\boxed{P^{*}_{\alpha\to\beta} = sin(2\theta)^2 sin\left(\frac{\Delta m^2 L}{4E}\right)^2}$$

Δm^2: frequency of oscillation

$\sin^2(2\theta)$: amplitude of oscillation

SBND MICROBOONE ICARUS



Reconstructed Neutrino Energy (GeV)

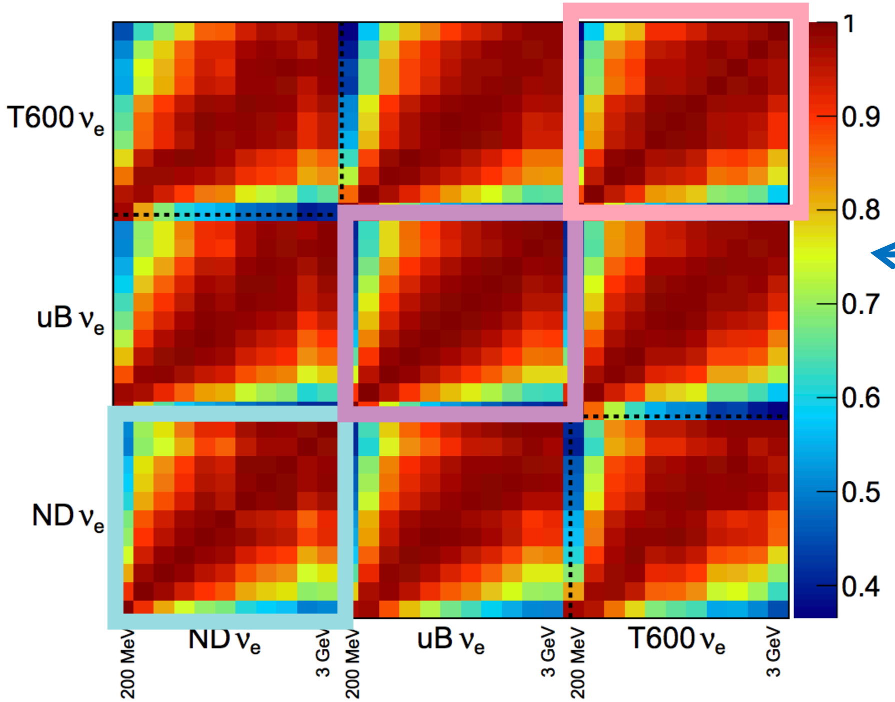
*Oscillation spectrum at Best Fit Point

[arXiv:1503.01520v1]

χ^2 Calculation for υ Oscillation Sensitivity

Sensitivity is calculated by computing a χ^2 surface in the $(\Delta m^2_{41}, \sin^2(2\theta))$ oscillation parameter plane:

$$\chi^2(\Delta m^2_{41}, \sin^2 2\theta) = \sum_{i,j} \left[N_i^{null} - N_i^{osc}(\Delta m^2_{41}, \sin^2 2\theta)\right] (E_{ij})^{-1} \left[N_j^{null} - N_j^{osc}(\Delta m^2_{41}, \sin^2 2\theta)\right]$$



Systematics uncertainties are computed in form of combined covariance matrix of the 3 detectors: SBND, MicroBooNE, ICARUS

[arXiv:1503.01520v1]

χ^2 Calculation for υ Oscillation Sensitivity

SBND — Neutrino energy spectrum

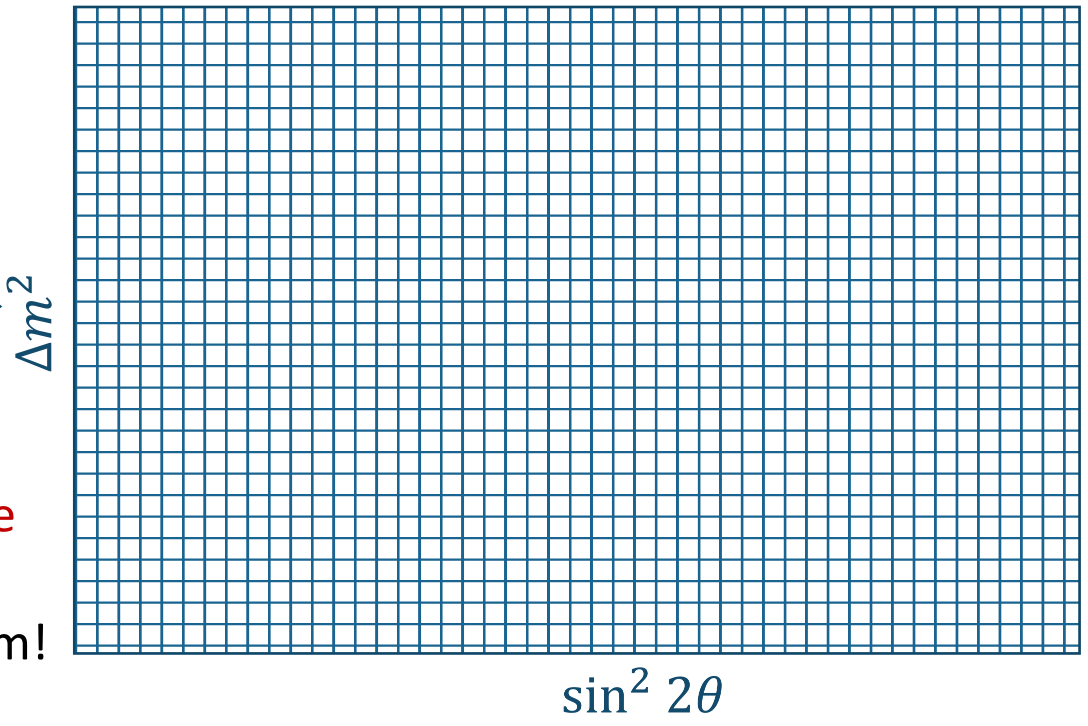
MicroBooNE — Neutrino energy spectrum

ICARUS — Neutrino energy spectrum

Covariance Matrix

Input

Perform χ^2 calculation at each **oscillation parameter space point** where we generate *N* number of toy experiments – Feldman Cousins procedure



Computationally expensive procedure but again highly parallel problem!

Output

ROOT Ntuples

n-dimensional contour plot

Accelerating SBNFit Feldman Cousins on HPC

SBND

Neutrino energy spectrum

MicroBooNE

Neutrino energy spectrum

ICARUS

Neutrino energy spectrum

Covariance Matrix

Input

Read once

HPC tools: MPI to distribute the work load to all ranks through memory

*MPI: Message Passing Interface

Perform χ^2 calculation at each **oscillation parameter space point per toy experiments per rank in a node** – Feldman Cousins procedure

Rank 0

Rank 1

Rank n

DIY Application

HPC tools: MPI to merge

Output

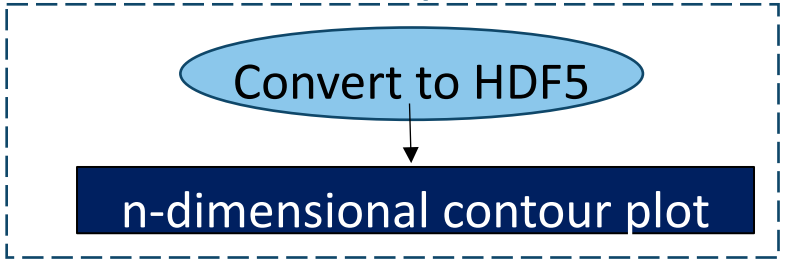
Convert to HDF5

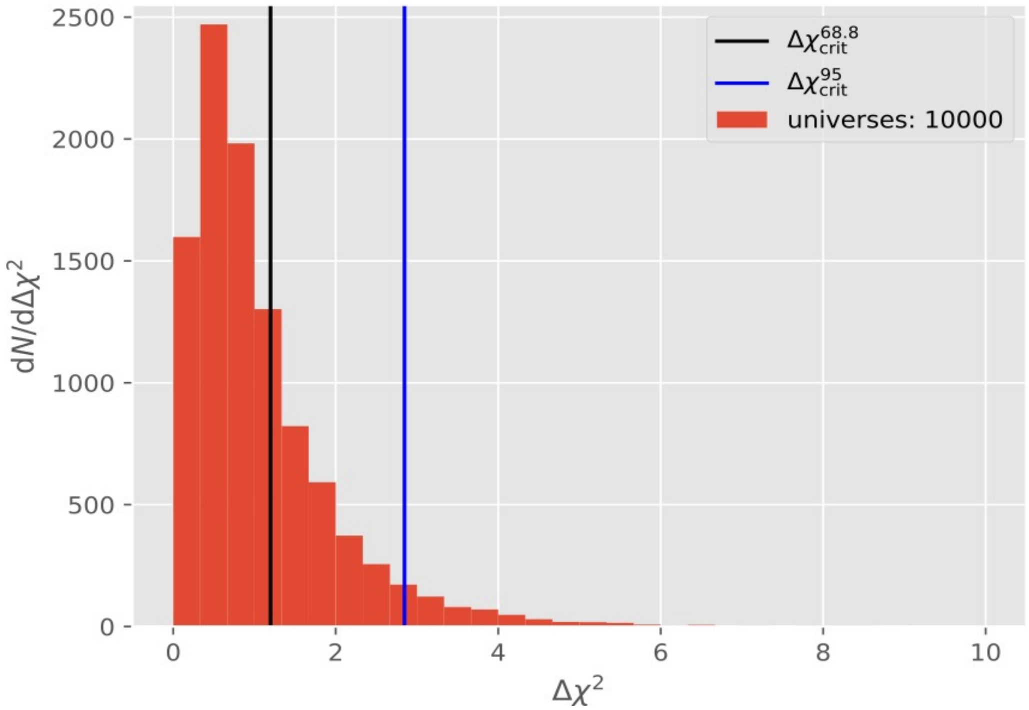
n-dimensional contour plot

Machine	Cori phase 1 (Haswell)	Cori phase 2 (KNL)
CPU	Intel Xeon E5-2698 v3	Intel Xeon Phi 7250
Clockspeed	2.3 GHz	1.4 GHz
Cores per node	32	68

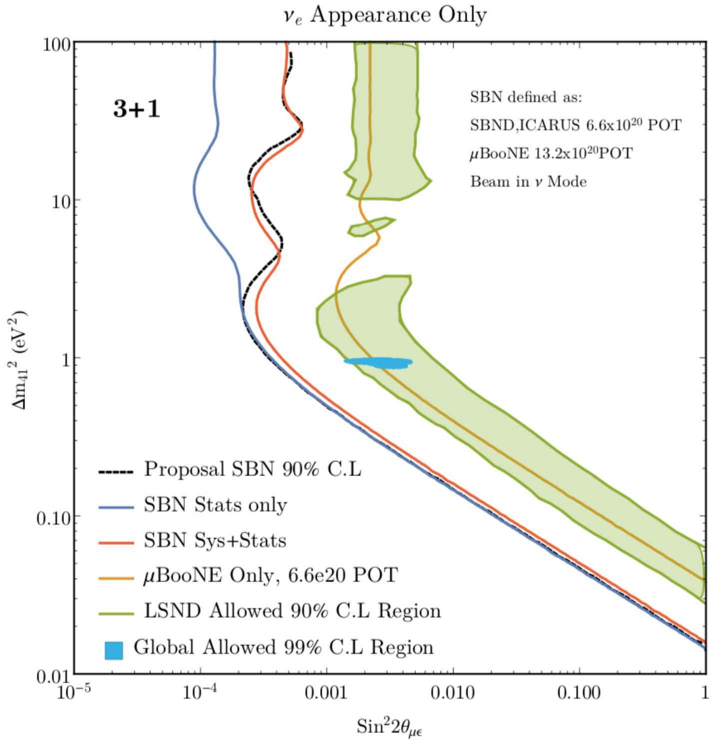
Accelerating SBNFit Feldman Cousins on HPC

Output





[arXiv:2002.07858v1]

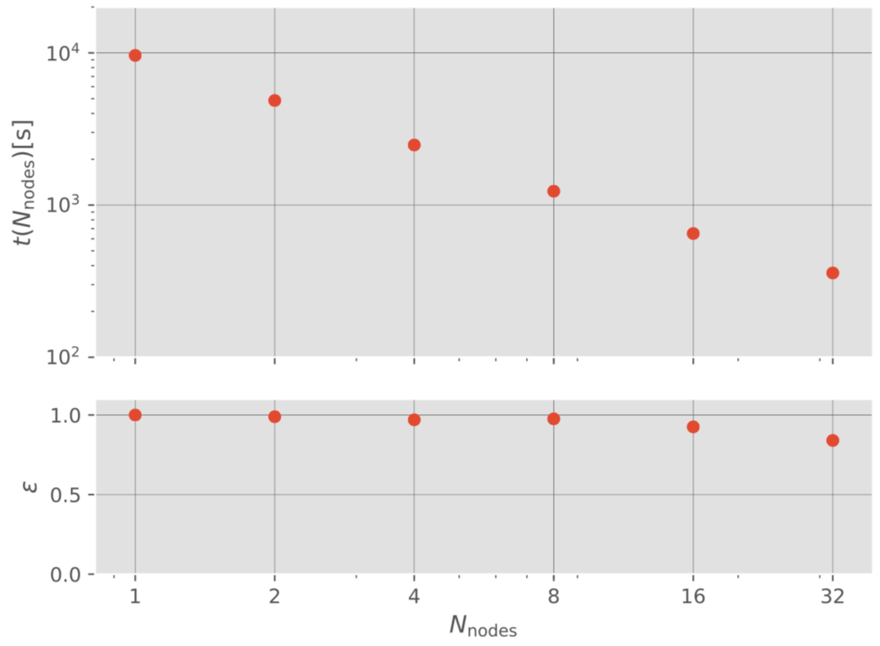


Phys. Rev . D 96, 055001 (2017)

"That sounds simple, so I can just run all my codes on those machines and get speedups!"

Scalability is Paramount!

- Scalability is important for parallel computing to be efficient.
- Most HEP codes use ROOT framework that does not scale well on HPC architectures. Need to identify algorithm alternatives and code bottlenecks.
- e.g. Restructure the codes to **read the input files only once using MPI** and use of **HDF5 as output format**. **Replace ROOT objects with types Eigen3** in linear algebra and vectorized calculation to compute χ^2. No longer memory-limited, only CPU-limited, allowing up to **350x speedups**.



Strong scaling measurements on Haswell nodes

[arXiv:2002.07858v1]

Allows probe to higher oscillation parameter dimensions, but with caveats, [see backup slide #29-30]

Summary

- As experiments grow in scale to explore **new physics with high precision** measurements and direct searches, it requires evolvement in our tools and software.
- In some regions, HPC facilities starting to play a major role, however we need an evolution of typical HEP workflows systems and the software itself to run efficiently.
  - **More challenges** going forward but also **more R&D opportunities**!
  - Computing division is an integral part of this.
- Other tools/venues to explore: **Quantum computing** holds the promise of remarkable new computational capabilities (more on the next talk)


THANK YOU

BACKUP SLIDES

χ^2 Test Statistics for Oscillation Analysis

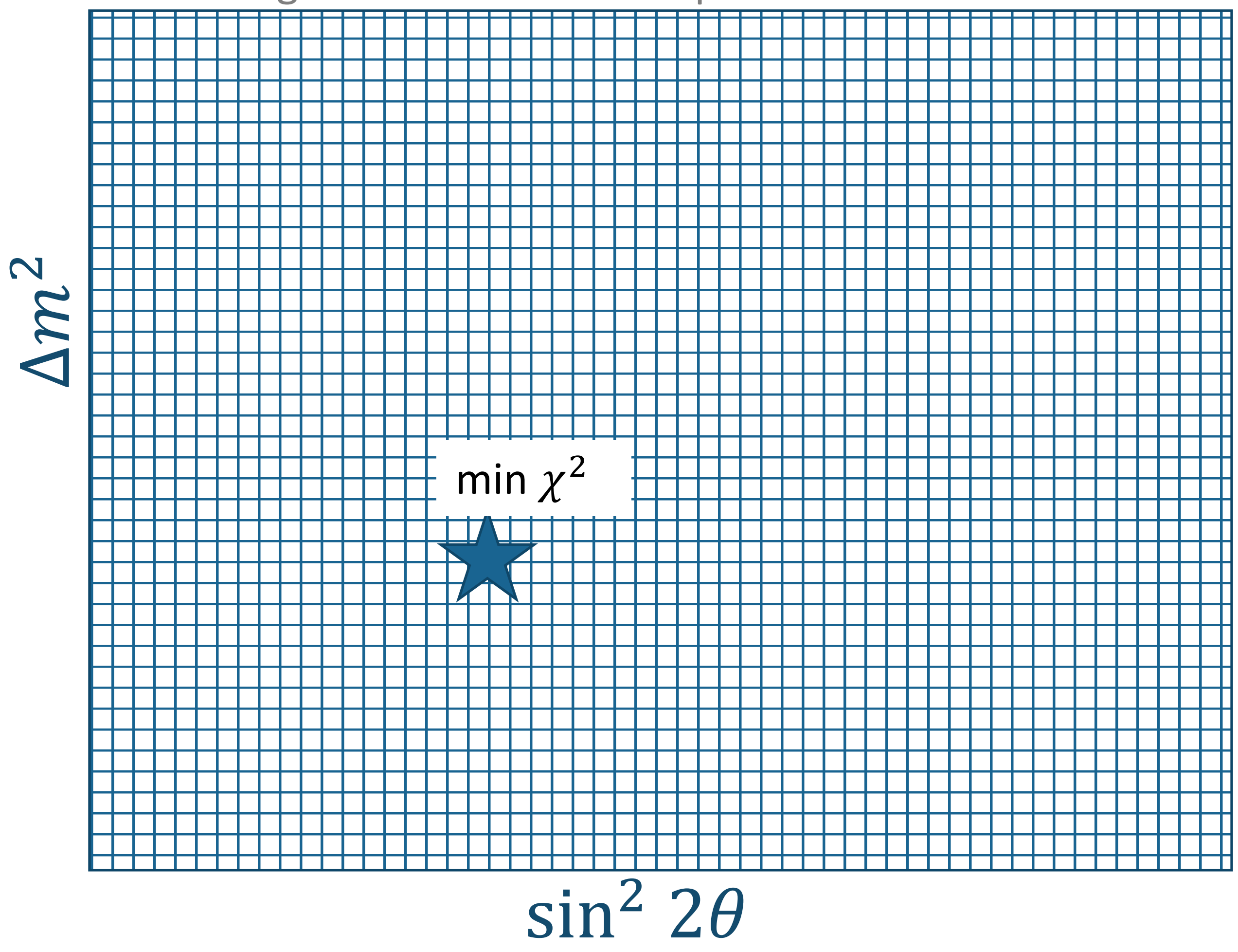
- Benchmarking:
  - SBNfit framework: one of the fitting framework used by SBN collaboration (Physics Group at Nevis Lab at Columbia University)
- Sensitivity is calculated by computing a χ^2 surface in the oscillation parameters (hyper)plane:

$$\chi^2(\Delta m^2_{i1}, U_{\alpha i}, \phi_{ij}) = \sum_{k=1}^{M}\sum_{l=1}^{M}\left[N_k^{\text{null}} - N_k^{\text{osc}}(\Delta m^2_{i1}, U_{\alpha i}, \phi_{ij})\right] E^{-1}_{kl} \left[N_l^{\text{null}} - N_l^{\text{osc}}(\Delta m^2_{i1}, U_{\alpha i}, \phi_{ij})\right]$$

3 scenarios:
3 parameters: 3N+1
7 parameters: 3N+2
12 parameters: 3N+3

Feldman cousins method

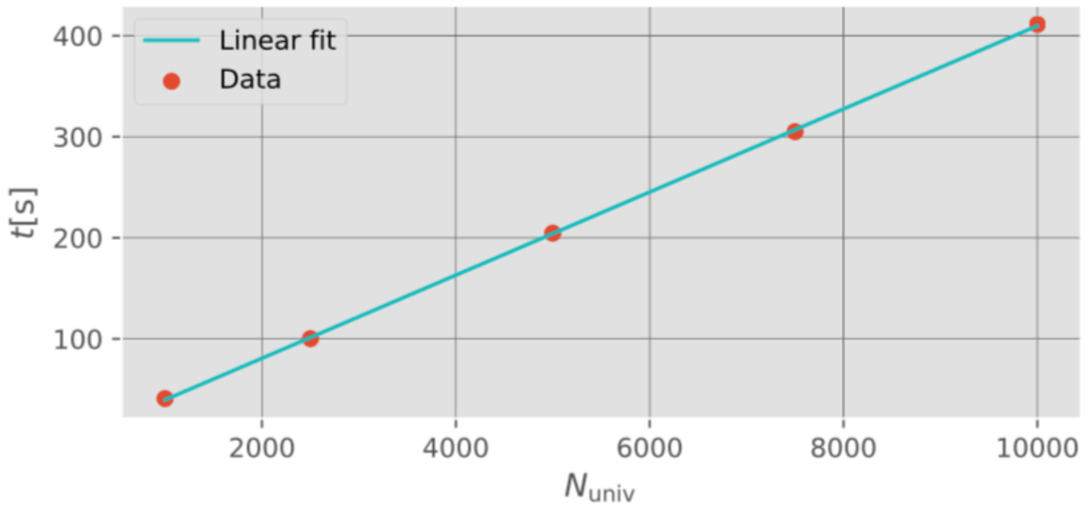
Define a grid dimension of the parameters scan



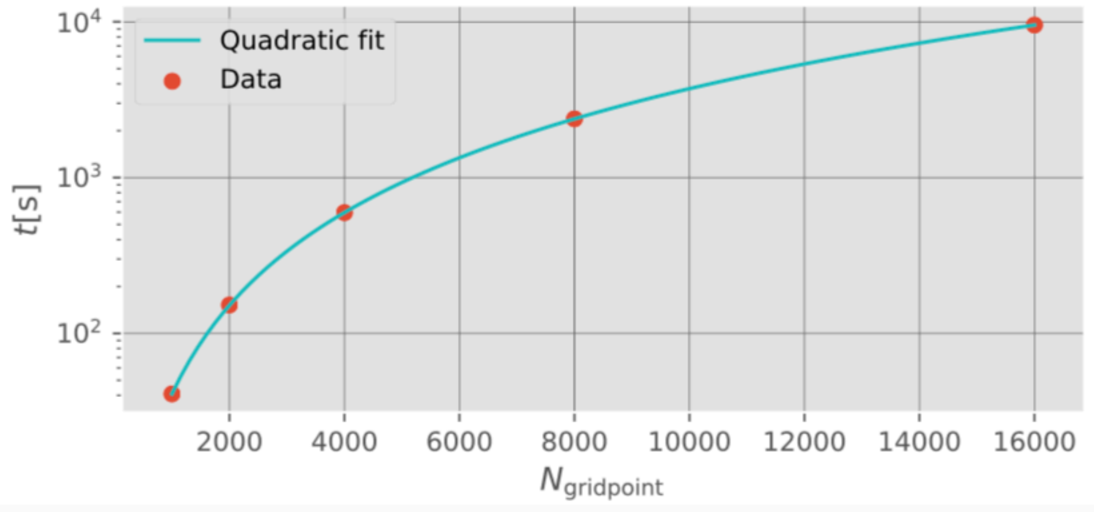
1. Given observed data spectrum D
2. Find the grid point with the minimum χ^2
3. Calculate $\Delta\chi^2$ at each parameter space point
4. Calculate the value of that 90% of experiments would be in by generating toy experiments.
5. Do this for 3σ (10^4 pseudo experiments/"universes") or 5σ ($\sim 10^8$ pseudo experiments/"universes") for more precision

Scaling on problem size

[arXiv:2002.07858v1]



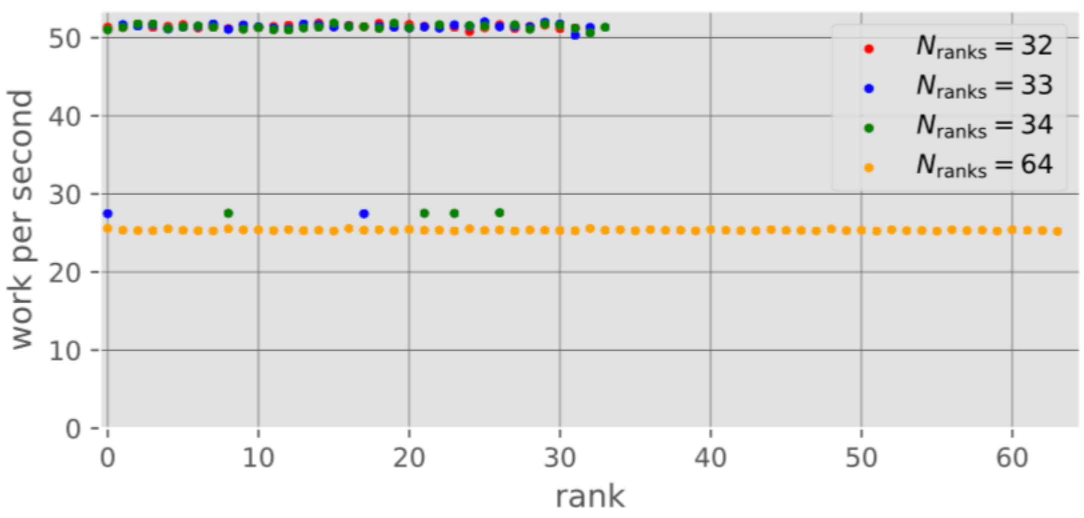
Scaling of the program run time with the number of universes, demonstrating a linear dependence on the number of pseudo-experiments.



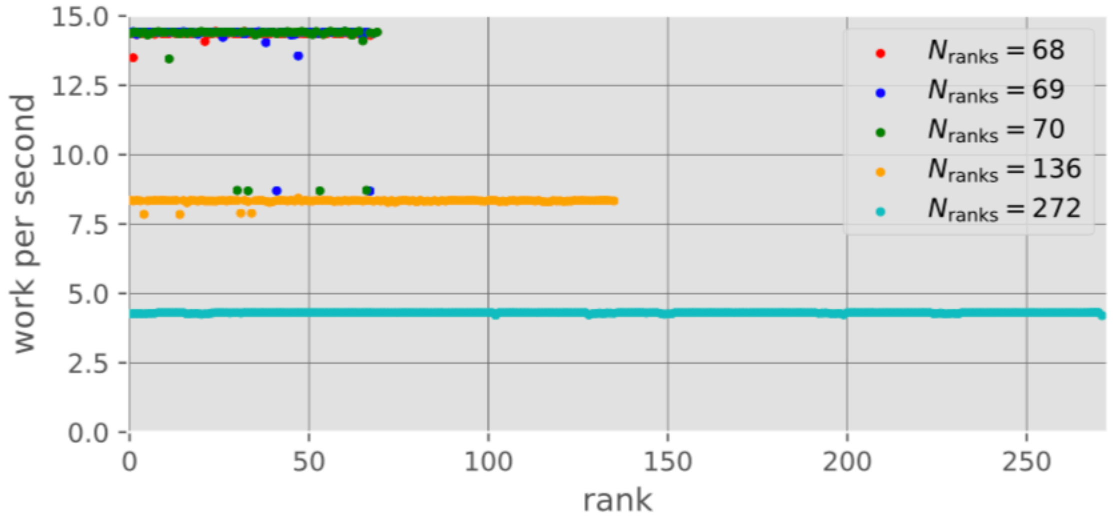
Scaling of the program run time with the number of grid points, demonstrating a quadratic dependence on the number of parameter space points.

Scaling on the single node

[arXiv:2002.07858v1]



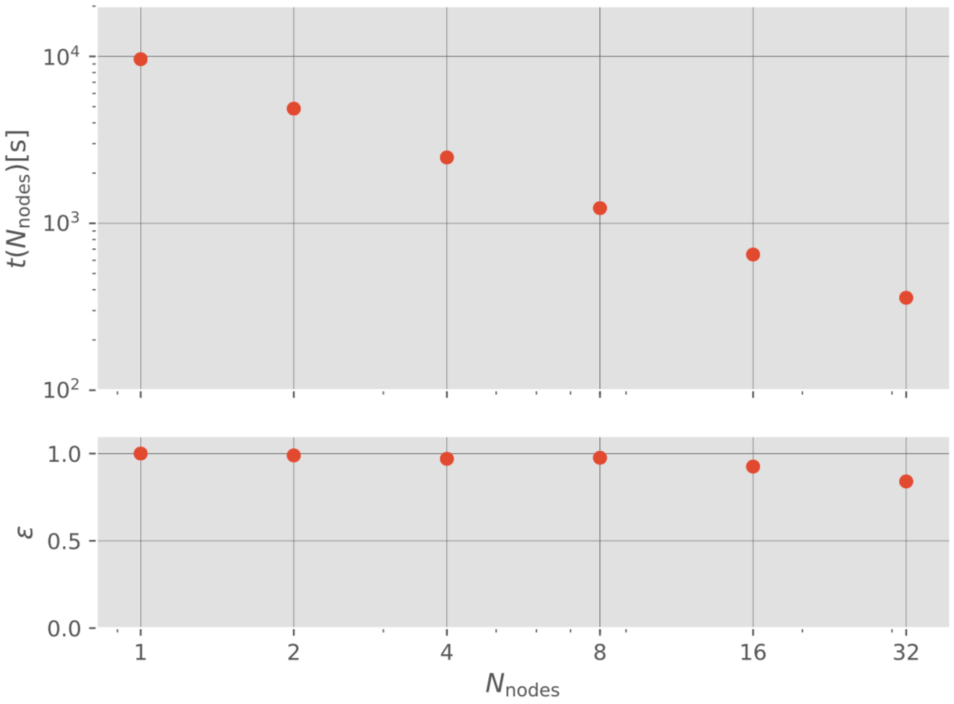
Measurement of the single Haswell node performance for a fixed problem size.



Measurement of the single KNL node performance for a fixed problem size

Scaling on multi-node

[arXiv:2002.07858v1]



Strong scaling measurements on Haswell nodes.
Generally good scaling up to the point where the amount of work per rank becomes relatively small.

Lesson Learned & Outlook

- The current method of scanning a regular grid becomes prohibitive after two or perhaps three dimensions.

- Exploring alternative techniques to this grid scan through the connection with the SciDAC FASTmath Institute.

- Explore changes necessary for evaluating the performance of GPU accelerators: change Eigen to Kokkos.

- These will allow the probe to higher-dimension parameter, such as measuring the sensitivities to 3N+2 paradigm